

AM



News Customers Products Downloads Pricing and Ordering Product Support About NetDirect Contact Us  
JDBC QuickStart J2EE Support



## JSQLConnect

JSQLConnect™ is a high performance, native type 4 JDBC driver providing comprehensive JDBC 3.0 and 2.0 support for Microsoft SQL Server™. JSQLConnect supports SQL Server 2000, 7.0, MSDE and 6.5. JDBC connectivity is provided for Java applications, Servlets, EJBs, JSPs and applets. JSQLConnect connects to SQL Server from any Java platform.

• Tec  
• FA  
• Bu

- [Latest New Features](#)
- [JDBC 3.0 Support](#)
- [Compare to Microsoft's JDBC Driver](#)
- [Product Evaluation Guide](#)
- [In Depth Technical Features](#)
- [J2EE Certification](#)
- [Product Feature Summary](#)
- [Performance Benchmark](#)
- [System Requirements](#)

### Latest New Features

Feature	Benefit	Reference
<b>NEW</b> Trusted Authentication	JSQLConnect can use integrated Windows Authentication mode to connect through a Microsoft Windows Windows® 2000 or NT® 4.0 user account. Provides increased security via your Windows domain controller. Avoids configuring and using additional user names and passwords and exposing them in configuration files.	<a href="#">Trusted Authentication</a>
<b>NEW</b> Distributed Transactions	JSQLConnect provides seamless integration with Microsoft® Distributed Transaction Coordinator. DTC uses proven transaction processing technology from Microsoft to support XA features such as the complete two-phase distributed commit protocol and the recovery of distributed transactions.	<a href="#">XA Connections</a>

### JDBC 3.0 Support

Feature	Benefit
Multiple Result Sets	If a statement returns multiple result sets the application may process any result set returned without having to close the other result sets.
Key Generation	When rows are inserted into the database the application has a standard way to automatically retrieve the row IDs generated.
Savepoints	The application may specify any number of savepoints in a transaction and specify a rollback to any savepoint.
Holdable Cursors	The application may control whether cursors are retained across a transaction commit or rollback.
Parameter Metadata	The application may obtain detailed meta data about every parameter required for a prepared statement or stored procedure.
Named Parameters	All parameters passed to stored procedures may now be referenced by name in addition to parameter number.
Statement Pooling	The application may specify statement pooling characteristics on data sources.
Blob/Clob	New API's are available to manipulate the contents of Blobs and Clobs.
Database Meta Data	Database meta data has been expanded to support the additional features introduced by the JDBC 3.0 specification.

### Product Feature Summary

- ✓ Full JDBC 3.0 Implementation
- ✓ SQL Server 2000, 7.0, 6.5 Compliant
- ✓ Sun Certified for J2EE™
- ✓ Sun Certified 100% Pure Java™
- ✓ Solaris, Linux, Windows Compliant
- ✓ JDK 1.4, 1.3 and 1.2 Compliant
- ✓ Optimized Native Protocol Performance
- ✓ Full Unicode Support
- ✓ Updateable Result Sets
- ✓ Scrolling Server Side Result Sets
- ✓ Scrolling Client Side Result Sets
- ✓ Full Range of Cursor Types
- ✓ True XA Support via MS-DTC
- ✓ Full BLOB and CLOB Support
- ✓ Optimistic and Pessimistic Concurrency
- ✓ Support for all Popular Java IDE's
- ✓ Complete Driver logging and tracing
- ✓ Full Timeout Support
- ✓ SQL Server Condition Codes in SQLExceptions
- ✓ NTEXT, NCHAR and NVARCHAR Support
- ✓ Unlimited Statements per Connection


## J2EE Support

- ✓ JNDI Data Sources
- ✓ Connection Pooling
- ✓ XA Distributed Transactions

## Product Evaluation Guide

In the market today there are a number of JDBC type 4 native drivers that provide JDBC access to Microsoft SQL Server™. This evaluation guide is provided by NetDirect to assist you in the evaluation of these products and to ensure that you choose the optimal product that supports the current and future requirements of your application.

All drivers should support the entire JDBC specification. In addition, we recommend reviewing the JDBC drivers you are currently evaluating against the points below before you commit to a specific product. Changing JDBC drivers once your application is completed is expensive in terms of both money and time. Code may need to be changed, all testing revalidated and components re-deployed. Even if you don't need all the JDBC features immediately, don't limit yourself now by a driver now that will not provide those features when you need them in the future.

Feature	Why It Is Important	How to Check
Pricing	Can you license every machine within your company for less than \$1500?	<a href="#">Check NetDirect Pricing</a>
Upgrades	Can you upgrade every machine within your company to the latest JDBC release for less than \$500?	<a href="#">NetDirect FastTrak Upgrades</a>
Redistribution	Can you redistribute the driver with your application for less than \$50 per machine?	<a href="#">Check NetDirect Pricing</a>
Support	Can you get immediate support from qualified engineers by phone or email for your JDBC questions?	<a href="#">NetDirect FastTrak Support</a>
Performance	How do other drivers compare on the benchmark performance test?	<a href="#">JDBC Performance Benchmark</a>
Technical Features	Do other drivers support all the technical features of JSQLEconnect?	<a href="#">NetDirect FastTrak Support</a>
J2EE Compliance	Is the driver vendor a J2EE partner? Will the vendor continue to upgrade the driver to support the latest J2EE specifications? For example, will the vendor support JDBC 3.0?	<a href="#">NetDirect News</a>
J2EE Certification	Is the driver certified by Sun for J2EE? J2EE certification ensures the driver will operate correctly with all J2EE compliant application servers.	Check for the  symbol at <a href="#">Sun's JDBC website</a> .

## In Depth Technical Features

Feature	Why It Is Important
Maximized Performance	By maximizing performance your users get more work done in a given time. A performance optimized driver also places less load on database resources and the JVM. Download NetDirect's <a href="#">JDBC performance benchmark suite</a> to compare the performance of any JDBC driver.
Connection Pooling	Connection pooling minimizes the number of physical connections required to the database. Connection times for the application are very fast. NetDirect connection pooling provides a wide range of datasource parameters to control and optimize connection pooling for your application.

Trusted Authentication	Trusted Authentication (integrated NT Authentication) provides increased security via your Windows domain controller. It avoids configuring and using additional user names and passwords and exposing them in configuration files.
Prepared Statements	Your application typically prepares statements once and executes them many times with different parameter values. The database should not be forced to re-evaluate the statement syntax every time it is executed since this significantly downgrades performance. <a href="#">More...</a>
Statement Pooling	If an application statement has already been seen by the driver its prepared state can be retrieved from driver cache, thereby avoiding additional load on the database. <a href="#">More...</a>
XA Distributed Transactions	Complete distributed transaction support must be available in the driver to ensure consistency of data across multiple database updates. The driver must interface with Microsoft® Distributed Transaction Coordinator which provides full distributed, two phase commit for SQL Server systems. Check that the driver ships with a mechanism to interface with DTC.
Cached Rowsets	Cached rowsets are invaluable feature for a number of architectural solutions. For example, when you need to pass a set of database rows from the business logic layer to the presentation layer which can present the data without requiring it's own database connection.

## Prepared Statements

### What are prepared statements?

Prepared statements are commonly employed in situations where the application will execute the same SQL statement many times. Only the values of the statement's parameters change from one execution of the statement to the next.

*The preparation of statements is an expensive process for the DBMS to execute, especially for complex SQL statements. For each statement preparation the DBMS must analyze the statement for syntactical correctness, validate column references and identify optimal access paths and execution plans.*

### How does JSQLConnect implement prepared statements

JSQLConnect uses SQL Server's sp\_prepare procedure to prepare the statement the first time it is prepared. Each time the statement is executed JSQLConnect calls the SQL Server sp\_execute procedure to execute the statement. Each call to sp\_execute simply passes the parameter values for this execution of the statement and does not incur the database overhead of re-preparing the statement.

## Statement and Connection Pooling

### Why does JSQLConnect's statement pooling improve performance and reduce DBMS load?

Statement pooling reduces the number of times the DBMS must prepare statements. The first time the application prepares a statement JSQLConnect will pass the statement to the DBMS for preparation. At this point JSQLConnect automatically caches the statement preparation.

Whenever the application prepares the statement again JSQLConnect will retrieve the statement from the statement cache rather than resubmitting it to the DBMS for preparation. This step thereby completely bypasses the overhead incurred in preparing the statement again and provides significant gains in JDBC performance and reduced load on the DBMS.

Statement caching is performed automatically and transparently by JSQLConnect. This means that programmers will continue to develop JDBC applications without requiring a knowledge of the operation of statement pooling.

### How does the JSQLConnect statement pool interoperate with pooled connections?

In many cases the application may be acquiring connections from a connection pool. For example J2EE EJB entity and session beans may use the application server's connection pool datasource. In this case it is common that the application acquires the connection only for the minimum time required to perform database updates and queries. As soon as the work is completed the connection will be returned to the connection pool.

Statement pools are retained when connections are returned to the connection pool. Therefore a newly acquired connection from the connection pool will usually have its statement pool already in place. In this case, even the application's first statement preparation on the connection will not require a DBMS statement preparation.

Statement pooling operates automatically on any connection, whether that connection was acquired from a non pooling data source, an application server pooling data source or a JSQLConnect connection pool.

## System Requirements

- Java Virtual Machine (JVM) 1.2 or higher.

- SQL Server 2000, 7.0 or 6.5.
- TCP/IP connectivity to SQL Server.
- JVM 1.4 or higher is required to use JSQLConnect 3.0 and the JDBC 3.0 features.

Sun, Sun Microsystems, Solaris, Personal Java, Java and JavaBeans are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Windows 95 and Windows NT are registered trademarks of Microsoft Corporation. JDataConnect is a trademark of NetDirect, Inc. All other brand names and products are trademarks or registered trademarks of their respective holders.

Copyright ©1998, 1999, 2000, 2001, 2002, 2003 NetDirect LLC